Journal of Computer and System Sciences 82 (2016) 333–346

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Meta-kernelization with structural parameters ☆,☆☆

Robert Ganian *, Friedrich Slivovsky, Stefan Szeider

Algorithms and Complexity Group, TU Wien, Vienna, Austria

A R T I C L E   I N F O

Article history:
Received 27 August 2014
Accepted 18 August 2015
Available online 10 September 2015

Keywords:
Parameterized complexity
Kernelization
Rank-width
Clique-width
Boolean-width
Monadic second-order logic
Modular decomposition

A B S T R A C T

Kernelization is a polynomial-time algorithm that reduces an instance of a parameterized problem to a decision-equivalent instance, the kernel, whose size is bounded by a function of the parameter. In this paper we present meta-theorems that provide polynomial kernels for large classes of graph problems parameterized by a structural parameter of the input graph. Let $\mathcal{C}$ be an arbitrary but fixed class of graphs of bounded rank-width (or, equivalently, of bounded clique-width). We define the $\mathcal{C}$*-cover number* of a graph to be the smallest number of modules its vertex set can be partitioned into, such that each module induces a subgraph that belongs to $\mathcal{C}$. We show that each decision problem on graphs which is expressible in Monadic Second Order (MSO) logic has a polynomial kernel with a linear number of vertices when parameterized by the $\mathcal{C}$-cover number. We provide similar results for MSO expressible optimization and modulo-counting problems.

1. Introduction

Kernelization is an algorithmic technique that has become the subject of a very active field in parameterized complexity, see, e.g., the references in [17,20,25,27,31,36]. Kernelization can be considered as a *preprocessing with performance guarantee* that reduces an instance of a parameterized problem in polynomial time to a decision-equivalent instance, the *kernel*, whose size is bounded by a function of the parameter alone; if the reduced instance is an instance of a different problem, then it is called a *bikernel*. Once a kernel or bikernel for a decidable problem has been obtained, the time required to solve the original instance is bounded by a function of the parameter and therefore independent of the input size. Consequently one aims at (bi)kernels that are as small as possible.

Every fixed-parameter tractable problem admits a kernel, but the size of the kernel can have an exponential or even non-elementary dependence on the parameter [19]. Thus research on kernelization is typically concerned with the question of whether a fixed-parameter tractable problem under consideration admits a small, and in particular a *polynomial*, kernel. For instance, the parameterized Minimum Vertex Cover problem (does a given graph have a vertex cover consisting of k vertices?) admits a polynomial kernel containing at most $2k$ vertices.

There are many fixed-parameter tractable problems for which no polynomial kernels are known. Recently, theoretical tools have been developed to provide strong theoretical evidence that certain fixed-parameter tractable problems do not admit polynomial kernels [3,22]. In particular, these techniques can be applied to a wide range of graph problems parameterized by treewidth and other width parameters such as clique-width, or rank-width (see e.g., [3,5]). Thus, in order to

☆ Research supported by the ERC (project COMPLEX REASON 239962) and by the FWF (project X-TRACT P26696).
☆☆ Parts of this paper appeared in preliminary and shortened form in the Proceedings of MFCS 2013.
* Corresponding author at: Algorithms and Complexity Group (186/1), TU Wien, Favoritenstrasse 9-11, A-1040 Vienna, Austria.
E-mail addresses: rganian@gmail.com (R. Ganian), fslivovsky@gmail.com (F. Slivovsky), stefan@szeider.net (S. Szeider).

http://dx.doi.org/10.1016/j.jcss.2015.08.003

get polynomial kernels, structural parameters have been suggested that are somewhat weaker than treewidth, including the *vertex cover number*, the *max-leaf number*, and the *neighborhood diversity* [18,29]. The general aim here is to find a parameter that admits a polynomial kernel for the given problem while being as general as possible.

We extend this line of research by using results from modular decompositions and rank-width to introduce new structural parameters for which large classes of problems have polynomial kernels. Specifically, we study the *$\mathcal{C}$-cover number* for graph classes $\mathcal{C}$ of bounded rank-width, or equivalently of bounded clique-width or bounded Boolean-width [7]. A graph G has $\mathcal{C}$-cover number k if k is the smallest number of modules the vertex set of G can be partitioned into, such that each module belongs to $\mathcal{C}$ (see Section 3 for exact definitions). For instance, taking $\mathcal{C}$ as the class of all graphs of rankwidth d, for $d = 1, 2, 3 \dots$ one obtains an infinite sequence of parameters that lie between the neighborhood diversity on the one side and rank-width on the other (see Fig. 1).

Our kernelization results take the shape of *algorithmic meta-theorems*, stated in terms of the evaluation of formulas of monadic second order logic (MSO) on graphs, similarly as in Courcelle's famous meta-theorem [10]. Monadic second order logic over graphs extends first order logic by variables that may range over sets of vertices (sometimes referred to as MSO_1 logic). Specifically, for an MSO formula φ, we consider the following problem, which we simply call an *MSO model checking* problem.

MSO-MC_φ
Instance: A graph G.
Question: Does $G \models \varphi$ hold?

Our first meta-theorem then states the following.

Theorem 1.1. *Let $\mathcal{C}$ be a graph class of bounded rank-width. Every MSO model checking problem, parameterized by the $\mathcal{C}$-cover number of the input graph, has a polynomial kernel with a linear number of vertices.*

Many NP-hard graph problems can be naturally expressed as MSO model checking problems.

Corollary 1.2. *Let $\mathcal{C}$ be a graph class of bounded rank-width. The following problems have polynomial kernels when parameterized by the $\mathcal{C}$-cover number of the input graph:* c-Coloring, c-Domatic Number, c-Partition into Trees, c-Clique Cover, c-Partition into Perfect Matchings, c-Covering by Complete Bipartite Subgraphs.

While MSO model checking problems already capture many important graph problems, there are some well-known optimization problems on graphs that cannot be captured in this way, such as Minimum Vertex Cover, Minimum Dominating Set, and Maximum Clique. Many such optimization graph problems can be equivalently stated as decision problems of the following form, which we call *MSO optimization problems*. Let $\varphi = \varphi(X)$ be an MSO formula with one free set variable X and $\diamond \in \{\leq, \geq\}$.

MSO-$\text{Opt}_\varphi^\diamond$
Instance: A graph G and an integer $r \in \mathbb{N}$.
Question: Is there a set $S \subseteq V(G)$ such that $G \models \varphi(S)$ and $|S| \diamond r$?

MSO optimization problems form a large fragment of the so-called *LinEMSO* problems[1] [2].

We establish the following result.

Theorem 1.3. *Let $\mathcal{C}$ be a graph class of bounded rank-width. Every MSO optimization problem, parameterized by the $\mathcal{C}$-cover number of the input graph, has a polynomial bikernel with a linear number of vertices.*

Corollary 1.4. *Let $\mathcal{C}$ be a graph class of bounded rank-width. The following problems have polynomial bikernels when parameterized by the $\mathcal{C}$-cover number of the input graph:* Minimum Dominating Set, Minimum Vertex Cover, Minimum Feedback Vertex Set, Maximum Independent Set, Maximum Clique, Longest Induced Path, Maximum Bipartite Subgraph, Minimum Connected Dominating Set.

In fact, the obtained bikernel is an instance of an "annotated" variant of the original MSO optimization problem [1]. Hence, Theorem 1.3 provides a polynomial kernel for an annotated version of the original MSO optimization problem.

Our third meta-theorem targets problems, where the objective is to compute the number of certain objects in the input modulo a constant. One very important class of such problems are *parity counting problems*, i.e., problems in the

[1] Our techniques apply to the entire class of LinEMSO problems, but we prefer focusing on MSO optimization problems to keep notation simple, while still capturing many concrete problems.